

REF. FILE COPY

2

Naval Medical Research Institute

Bethesda, MD 20814-6055

NMFI 88-6

AUGUST 1988



PERFORMANCE ASSESSMENT BATTERY SOFTWARE

AD-A199 828

DTIC
ELECTE
OCT 06 1988
S D

John Schrot and John R. Thomas

Approved for public release;
distribution is unlimited

Naval Medical Research
and Development Command
Bethesda, Maryland 20814-5044

Department of the Navy
Naval Medical Command
Washington, D.C. 20372-5210

88 10 5 18 9

NOTICES

The opinions and assertions contained herein are the private ones of the writer and are not to be construed as official or reflecting the views of the naval service at large.

When U.S. Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Please do not request copies of this report from the Naval Medical Research Institute. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22161

Federal Government agencies and their contractors registered with the Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center
Cameron Station
Alexandria, Virginia 22304-6145

TECHNICAL REVIEW AND APPROVAL

NMRI 88-8

The experiments reported herein were conducted according to the principles set forth in the current edition of the "Guide for the Care and Use of Laboratory Animals," Institute of Laboratory Animal Resources, National Research Council.

This technical report has been reviewed by the NMRI scientific and public affairs staff and is approved for publication. It is releasable to the National Technical Information Service where it will be available to the general public, including foreign nations.

K. SORENSEN, CAPT MC, USN

Commanding Officer
Naval Medical Research Institute

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE				
1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b RESTRICTIVE MARKINGS	
2a SECURITY CLASSIFICATION AUTHORITY			3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited	
2b DECLASSIFICATION/DOWNGRADING SCHEDULE				
4 PERFORMING ORGANIZATION REPORT NUMBER(S) NMRI 88-8			5 MONITORING ORGANIZATION REPORT NUMBER(S)	
6a NAME OF PERFORMING ORGANIZATION NAVMEDRSCHINSTITUTE		6b OFFICE SYMBOL (if applicable)	7a NAME OF MONITORING ORGANIZATION NAVAL MEDICAL COMMAND	
6c ADDRESS (City, State, and ZIP Code) NMCNCR BETHESDA, MARYLAND 20814-5055			7b ADDRESS (City, State, and ZIP Code) DEPARTMENT OF THE NAVY WASHINGTON, DC 20372-5120	
8a NAME OF FUNDING/SPONSORING ORGANIZATION USAMRDC		8b OFFICE SYMBOL (if applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c ADDRESS (City, State, and ZIP Code) Fort Detrick Frederick, MD 21071			10 SOURCE OF FUNDING NUMBERS	
			PROGRAM ELEMENT NO 63764A 63706N	PROJECT NO 3M463764 M0095
			TASK NO D994AB081-1 004.1000	WORK UNIT ACCESSION NO DA 303502 DN246556
11 TITLE (Include Security Classification) PERFORMANCE ASSESSMENT BATTERY SOFTWARE				
12. PERSONAL AUTHOR(S) JOHN SCHROT AND JOHN R. THOMAS				
13a. TYPE OF REPORT TECHNICAL		13b. TIME COVERED FROM Oct 84 to Oct 86	14. DATE OF REPORT (Year, Month, Day) August 1, 1988	15. PAGE COUNT
16. SUPPLEMENTARY NOTATION				
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP		
			Performance Assessment, NMRI-PAB Battery Tests	
19 ABSTRACT (Continue on reverse if necessary and identify by block number) The computer software for the Naval Medical Research Institute Performance Assessment Battery (NMRI-PAB) is provided. A previous report presented a detailed description and specification of the assessment battery, as well as the rationale for its implementation. The present report includes the complete computer source code for all components of the NMRI-PAB, written in BASIC programming language.				
20 DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS			21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL R. E. H. t, Command Editor, NMRI			22b TELEPHONE (include Area Code) (202) 295-0198	22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

ACKNOWLEDGEMENTS

This report was supported by the Naval Medical Research and Development Command Research and Technology Work Units, 63706N M00095.004.1008 and 63764A 3M463764B995.AB.081-1. The opinions and assertions contained herein are those of the authors and are not to be construed as official or reflecting the views of the Navy Department or the Naval Service at large.

REVISION	
REF. NO.	✓
DATE	11
TIME	11
BY	
FOR	
REMARKS	
A-1	



TABLE OF CONTENTS

Introduction	4
NMRI-PAB Language Information	5
NMRI-PAB System Features	6
Executive Program	10
Matching to Sample Test	12
Stroop Test	16
Grammatical Reasoning Test	20
Manikin Test	24
Numerical Memory Test	30
Pattern Comparison Test	35
Repeated Acquisition Test	39
Visual Scanning Test	43
Common Program	46

INTRODUCTION

In order to measure the effects of operational environments on cognitive performance, the Naval Medical Research Institute Performance Assessment Battery (NMRI-PAB) was developed. The rationale for the development and implementation of the NMRI-PAB and a detailed description and specification of the Executive program and the eight tests that constitute the assessment system were presented in a previous report, 'Naval Medical Research Institute Performance Assessment Battery (NMRI-PAB): Performance Assessment Battery Documentation.' The present report provides the complete computer source code for the NMRI-PAB. The source code includes the Executive program and the eight component tests of the battery. The Executive program organizes and controls the individual tests of the NMRI-PAB. The eight component tests are: the Matching to Sample, the Stroop, the Grammatical Reasoning, the Manikin, the Numerical Memory, the Pattern Comparison, the Repeated Acquisition, and the Visual Scanning tests. Each of these components is designed to be compiled as a separate executable program. The Executive program 'calls' each individual test as it is required.

NMRI-PAB LANGUAGE INFORMATION

The present software is written in the BASIC programming language. The source code has been designed to operate with and be compiled by Microsoft QuickBASIC, Version 3. The source code was originally written in BASICA and GW-BASIC and later translated to QuickBASIC. Relatively minor code changes should allow the software to operate with a variety of BASIC interpreters and compilers. The QuickBASIC Version 3 language requires an IBM PC/AT or compatible computer. The source code is designed to function with the Microsoft Disk Operating System (MS-DOS or PC-DOS), Version 2.0 or later.

NMRI-PAB SYSTEM FEATURES

The source code generally follows a similar format for all eight of the battery tests. Certain common programming features are found in all of the tests and are briefly outlined here.

At the beginning of the source code listing of each test the "\$INCLUDE" metacommand instructs the program to switch to a program called "COMMON.BAS". The Executive program initially obtains the subject's name, date, filename, etc., and later passes that information to the common program variables. Each individual test can then retrieve that information via the common variables. This allows specific information (subject's name, date, etc.) to be appended to all data files and to control particular programming functions in certain individual tests.

All timing aspects are controlled by the software clock that is part of the Disk Operating System, which in turn is dependent upon the computer hardware clock. The most accurate timing that is possible through the software is at a resolution of hundreds of a second. Near the end of the source code listing for each test there are two routines concerned with timing aspects. One of these routines reads the software clock and retrieves the time as hours : minutes : seconds : hundreds of seconds. This routine uses a Disk Operating System call to obtain the time. The other

timing routine converts hours : minutes : seconds format to seconds only. These timing functions are used repeatedly throughout the component tests, for example, to time the duration of a test.

At the start of each test a 20 second inter-test-interval is programmed using the above timing function. The interval consists of a 14 second blank screen followed by a 6 second "traffic light." During the "traffic light" the screen is red for 2 seconds, yellow for 2 seconds, and then green for 2 seconds.

All tests store information about each trial in a data array that is updated during each trial. At the end of each test a data file is opened, given the name that was passed from the common program (from the Executive program) along with a three letter extension that is unique to each test, and all of the data is written to the data file.

In all of the tests, an attempt has been made to insure that only one response can occur at a time (two response buttons can not be pressed at the same time) and that a button must be released before it can be sequentially pressed again (holding a button down throughout several program events will not be recorded as multiple responses).

The source code assumes that all response inputs to the system are from a parallel digital input/output device such as the Intel 8255 parallel port. The addresses of the parallel device in the present code are those of the Scientific Solutions

Lab Master board. Changes in these addresses would allow the source code to be used with other devices. The code can also be easily modified such that all INP functions (returns byte read from the input/output port device) are replaced with the INKEY\$ function to use keyboard inputs as responses.

The source code of the tests present video images on the screen in either Screen Mode 0 (text mode), designed for a 40 by 25 text format or in Screen Mode 1, designed for 320 by 200 pixel medium-resolution graphics. All screens are cleared and Screen Mode 0 is reinstated at the termination of each test. The video aspects of the source code were designed to operate on either a Zenith 150 PC computer or on a system using the Sigma Designs Color 400-SH-512 video adapter. Other video display systems can be used by supplying the appropriate values for the hardware constants ("CRT", "BLANK", and "UNBLANK") near the beginning of each test code listing.

An attempt was made to optimize the code to operate as rapidly and efficiently as possible, particularly in regards to performance relevant variables. Special emphasis was placed on obtaining the most rapid time measures following the occurrence of each recorded response and with the relationship of responses to video presentations.

For the tests to operate at the most desirable speed for performance measurement, they should be compiled for speed enhancement, where ever possible. The authors compile the source code of each of the individual NMRI-PAB programs using three

arguments to the QuickBASIC compile command: /q produces a program that runs as quickly as possible; /g allows the video screen to be updated as fast as possible; and /l loads the user library file. The source code for the NMRI-PAB is designed to use the QuickBASIC supplied user library, USERLIB, which must be available to the individual programs at run-time.

```

      Executive Program
      EXEC.BAS
      07-29-88

```

```

REM $INCLUDE: 'COMMON.BAS'
ADDRESS=1808: OUT ADDRESS+15,146 ' Setup Techmar board
OUT ADDRESS+14,0 ' Turn all leds off on panel
DIM TN$(10)
NXT=1:SCREEN 2:SCREEN 0:COLOR 7,1:CLS
LOCATE 5,37:PRINT "NMRI PAB"
LOCATE 6,36:PRINT "Written by:"
LOCATE 7,29:PRINT "J.R. Thomas and J. Schrot"
LOCATE 8,25:PRINT "Environmental Medicine Department"
LOCATE 9,26:PRINT "Naval Medical Research Institute"
LOCATE 10,31:PRINT "Bethesda, MD. 20814"
PRINT:PRINT:PRINT
INPUT "Enter subject's name: ",SN$
PRINT
INPUT "Enter date (DD-MM-YY): ",CD$
PRINT
INPUT "Enter file name (drive:filename): ",F$
PRINT
INPUT "Enter random number seed (-32768 to 32767): "
      RANDNUMB
RANDOMIZE(RANDNUMB)
PRINT
1190 INPUT "Enter repeated acquisition sequence number (1 to 24): "
      ,SEQNUM
IF SEQNUM < 1 OR SEQNUM > 24 THEN 1190
PRINT:PRINT
PRINT "The standard test sequence is Numerical Memory,
      Pattern Comparison,"
PRINT "Grammatical Reasoning, Matching-to-Sample,
      Visual Scanning"
PRINT "Manikin, Repeated Acquisition and Stroop."
PRINT
INPUT "Do you wish to run the standard sequence? (Y or N) "
      ,SS$
IF SS$="N" OR SS$="n" THEN 1320
T$(1)="NUMPAN":T$(2)="PATPAN":T$(3)="GRAMPAN"
T$(4)="MATPAN":T$(5)="VISPAN":T$(6)="MANPAN"
T$(7)="RAPAN":T$(8)="STROPPAN":N=8
GOTO 1590

1320 ' Menu to construct test sequence
CLS

```

```

1335 LOCATE 10,5:INPUT "Type number of tests to be administered
      ( 1 to 8 ) : ",N
      IF N<1 OR N>8 THEN 1335
      LOCATE 10,5:PRINT "
      .

      LOCATE 5,30:PRINT "TEST SEQUENCE SELECTION"
      LOCATE 7,34:PRINT "TESTS AVAILABLE"
      PRINT
      LOCATE 9,22:PRINT "TEST NAME"
      LOCATE 9,55:PRINT "FILE NAME"
      PRINT
      PRINT "          1.  Grammatical Reasoning
      GRAMPAN"
      PRINT "          2.  Visual Scanning
      VISPAN"
      PRINT "          3.  Manikin
      MANPAN"
      PRINT "          4.  Matching-to-Sample
      MATPAN"
      PRINT "          5.  Pattern Comparison
      PATPAN"
      PRINT "          6.  Repeated Acquisition
      RAPAN"
      PRINT "          7.  Numerical Memory
      NUMPAN"
      PRINT "          8.  Stroop
      STROPPAN"
      TN$(1)="1st":TN$(2)="2nd":TN$(3)="3rd":TN$(4)="4th":
      TN$(5)="5th":TN$(6)="6th":TN$(7)="7th":TN$(8)="8th"
      FOR J=1 TO N
          LOCATE 20,5:PRINT "
          .

          LOCATE 20,5:PRINT "Enter the number of the ";:
          PRINT TN$(J)
          LOCATE 20,33:INPUT "test: ",TN
          GOSUB 1620
      NEXT J

      ' Chain tests
1590 COLOR 7,0:CLS:LOCATE 1,1,0
      CHAIN T$(NXT)

1620 ' Assign tests in sequence
      IF TN=1 THEN T$(J)="GRAMPAN":GOTO 1710
      IF TN=2 THEN T$(J)="VISPAN":GOTO 1710
      IF TN=3 THEN T$(J)="MANPAN":GOTO 1710
      IF TN=4 THEN T$(J)="MATPAN":GOTO 1710
      IF TN=5 THEN T$(J)="PATPAN":GOTO 1710
      IF TN=6 THEN T$(J)="RAPAN":GOTO 1710
      IF TN=7 THEN T$(J)="NUMPAN":GOTO 1710
      IF TN=8 THEN T$(J)="STROPPAN"
1710 RETURN

```

```

'           Matching-to-Sample
'           MATPAN.BAS
'           07-28-88

```

```

REM $INCLUDE: 'COMMON.BAS'
'CRT=986: BLANK=0: UNBLANK=1: OUT 984,2 ' Z150
CRT=728: BLANK=7: UNBLANK=15 ' Color 400-SH
ADDRESS=1808 ' Techmar board
OUT ADDRESS+15,146 ' Ports A + B input, C output
OUT ADDRESS+14,0 ' All leds off

DIM SETREG%(7), GETREG%(7)
AXREG%=0: CXREG%=2: DXREG%=3

LOCATE 1,1,0
DIM A(20),D(20),CORRECT(100),ERRORS(100),DAT(31,32),ZSEC(8)

Z=7:GOSUB 2650
1210 Z=8:GOSUB 2650
    IF ZSEC(8) < ZSEC(7)+14 THEN 1210
    COLOR 7,4:CLS
1240 Z=8:GOSUB 2650
    IF ZSEC(8) < ZSEC(7)+16 THEN 1240
    COLOR 7,6:CLS
1270 Z=8:GOSUB 2650
    IF ZSEC(8) < ZSEC(7)+18 THEN 1270
    COLOR 7,2:CLS
1300 Z=8:GOSUB 2650
    IF ZSEC(8) < ZSEC(7)+20 THEN 1300
    COLOR 7,0:CLS
    OUT ADDRESS+14,4 ' Panel led on
    TRIALS=30:TOTC=0:TOTE=0

'Get start of session time
Z=1:GOSUB 2650
FOR T=1 TO TRIALS
    'Check for end of session
    IF T=1 THEN GOTO 1430
    Z=2:GOSUB 2650
    IF ZSEC(2) >= ZSEC(1) + 300 THEN NTC=T-1: GOTO 1920

1430 CLS
    S=INT(RND*2)+1
    ' Generate order of 16 cells in sample
    and correct matrix
    FOR J=1 TO 16
        A=INT(RND*10)+1

```

```

        IF A > 5 THEN A=1 ELSE A=2
        A(J)=A: D(J)=A: DAT(T,(J+14))=A
    NEXT J
    ' Change one cell in the S- comparison matrix
    A=INT(RND*16)+1
    IF D(A)=1 THEN D(A)=2 ELSE D(A)=1
    DAT(T,31)=A
    CLS: SCREEN 1: COLOR 0,0

    ' Draw sample stimulus
1580 RS=INP(ADDRESS+13): IF RS<>255 THEN 1580
    OUT CRT,BLANK
    GOSUB 2130: GOSUB 2360 ' Draw sample stimulus
    OUT CRT, UNBLANK
1620 RS=INP(ADDRESS+13): IF RS=255 THEN 1630 ELSE 1620
1630 K=0:GOSUB 2570 ' Get initial IRT value
    Z=3:GOSUB 2650
1650 Z=4:RS=INP(ADDRESS+13):IF RS=239 THEN 1690
    GOSUB 2650: IF ZSEC(4) >= ZSEC(3) + 60 THEN 1690
    Z=2:GOSUB 2650:IF ZSEC(2) >= ZSEC(1)+300 THEN NTC=T-1:
    GOTO 1920
    GOTO 1650
1690 K=4:GOSUB 2570 ' Store time when screen is advanced
    OUT CRT,BLANK ' Clear screen

    ' Draw comparison stimuli
1730 RS=INP(ADDRESS+13):IF RS<>255 THEN 1730
    GOSUB 2180 ' Draw two comparison stimuli
    OUT CRT,UNBLANK
1760 RS=INP(ADDRESS+13):IF RS=255 THEN 1770 ELSE 1760
1770 Z=2:RS=INP(ADDRESS+13)
    IF RS=247 OR RS=251 THEN GOTO 1810 ELSE GOSUB 2650
    IF ZSEC(2) >= ZSEC(1) + 300 THEN NTC=T-1:GOTO 1920
    GOTO 1770
1810 K=8: GOSUB 2570
    IF S=1 AND RS=247 THEN
        CORRECT(T)=CORRECT(T)+1:DAT(T,13)=1:TOTC=TOTC+1
    IF S=1 AND RS=251 THEN
        ERRORS(T)=ERRORS(T)+1:DAT(T,13)=2:TOTE=TOTE+1
    IF S=2 AND RS=251 THEN
        CORRECT(T)=CORRECT(T)+1:DAT(T,13)=1:TOTC=TOTC+1
    IF S=2 AND RS=247 THEN
        ERRORS(T)=ERRORS(T)+1:DAT(T,13)=2:TOTE=TOTE+1

NEXT T

NTC=TRIALS
1920 SCREEN 2: SCREEN 0:COLOR 7,0:CLS:LOCATE 1,1,0
    COLUMNS 31
    OPEN "O",1,F$+".MAT"
    PRINT#1,SN$;"",CD$
    PRINT#1,NTC,COLUMNS

```

```

FOR TR=1 TO NTC
  PRINT#1,DAT(TR,1);DAT(TR,2);DAT(TR,3);
  DAT(TR,4);DAT(TR,5);DAT(TR,6);
  PRINT#1,DAT(TR,7);DAT(TR,8);DAT(TR,9);
  DAT(TR,10);DAT(TR,11);DAT(TR,12);
  PRINT#1,DAT(TR,13);DAT(TR,14);DAT(TR,15);
  DAT(TR,16);DAT(TR,17);DAT(TR,18);
  PRINT#1,DAT(TR,19);DAT(TR,20);
  PRINT#1,DAT(TR,21);DAT(TR,22);DAT(TR,23);
  DAT(TR,24);DAT(TR,25);DAT(TR,26);
  PRINT#1,DAT(TR,27);DAT(TR,28);DAT(TR,29);
  DAT(TR,30);DAT(TR,31)
NEXT TR: CLOSE
NXT=NXT+1
IF NXT>N THEN PRINT "END OF SESSION":
  OUT ADDRESS+14,0:END:LOCATE 1,1,1
LOCATE 1,1,0
CHAIN T*(NXT)

' Draw sample stimulus
2130 CLS
Y=80:X=125
GOSUB 2270
RETURN
' Draw two comparison stimuli
2180 CLS
X=65
GOSUB 2270: IF S=1 THEN GOSUB 2360 ELSE GOSUB 2470
X=185
GOSUB 2270: IF S=1 THEN GOSUB 2470 ELSE GOSUB 2360
IF S=1 THEN DAT(T,14)=1 ELSE DAT(T,14)=2
RETURN

' Draw matrix
2270 FOR I = Y TO Y + 40 STEP 10
  LINE (X,I)-(X+60,I),3
NEXT I
FOR I = X TO X + 60 STEP 15
  LINE (I,Y)-(I,Y+40),3
NEXT I
RETURN

' Paint Correct Matrix
2360 J=1
FOR E=0 TO 30 STEP 10
  FOR B 0 TO 45 STEP 15
    C3=A(J)
    PAINT (X+5*B,Y+5+E),C3,3
    J=J+1
  NEXT B
NEXT E
RETURN

```



```

      ' Paint Incorrect Matrix
2470 J=1
      FOR E=0 TO 30 STEP 10
        FOR B=0 TO 45 STEP 15
          C3=D(J)
          PAINT (X+5+B,Y+5+E),C3,3
          J=J+1
        NEXT B
      NEXT E
      RETURN

2570 ' Read clock and store time in array
      SETREGZ(AXREGZ)=&H2C00
      CALL INT86(&H21,VARPTR(SETREGZ(0)),VARPTR(GETREGZ(0)))
      HM=GETREGZ(CXREGZ): SH=GETREGZ(DXREGZ)
      DAT(T,(K+1))=HM \ 256: DAT(T,(K+2))=HM MOD 256
      DAT(T,(K+3))=SH \ 256: DAT(T,(K+4))=SH MOD 256
      RETURN

2650 ' Convert Hr:Min:Sec to Sec and test for end
      ZTIME$=TIME$
      ZS=VAL(RIGHT$(ZTIME$,2))
      ZM=VAL(MID$(ZTIME$,4,2))
      ZH=VAL(LEFT$(ZTIME$,2))
      ZM1= (ZH * 60) + ZM
      ZSEC(Z)= (ZM1* 60) + ZS
      RETURN

```

```

      Stroop
      STROPPAN.BAS
      07-28-88

REM #INCLUDE: 'COMMON.BAS'
ADDRESS=1808: OUT ADDRESS+15,146
OUT ADDRESS+14,0

SCREEN 2:SCREEN 0:COLOR 7,0:CLS:LOCATE 1,1,0

'CRT=986: BLANK=0: UNBLANK=1: OUT 984,2 ' Z150
CRT=728: BLANK=7: UNBLANK=15 ' Color 400-SH

DIM SETREG%(7), GETREG%(7)
AXREG%=0: CXREG%=2: DXREG%=3

DIM ERRORS(100),CORRECT(100),DAT(63,12),ZSEC(6),
    A(10),DIST(63)

S=5:GOSUB 2650
TRIALS=45

FOR I=1 TO TRIALS
1140   B=INT(RND*9)+1
       IF A(B) >4 GOTO 1140
       DIST(I)=B
       A(B)=A(B)+1
NEXT I

1190 S=6:GOSUB 2650
     IF ZSEC(6) < ZSEC(5)+14 THEN 1190
     COLOR 7,4:CLS ' Screen Red
1220 S=6:GOSUB 2650
     IF ZSEC(6) < ZSEC(5)+16 THEN 1220
     COLOR 7,6:CLS ' Screen Yellow
1250 S=6:GOSUB 2650
     IF ZSEC(6) < ZSEC(5)+18 THEN 1250
     COLOR 7,2:CLS ' Screen Green
1280 S=6:GOSUB 2650
     IF ZSEC(6) < ZSEC(5)+20 THEN 1280

OUT ADDRESS+14,4
' TEST=1: Respond to words with colors irrelevant
' TEST=2: Respond to colors with words irrelevant
' TEST=3: Respond to words only (reaction time test
          - words all in white)

```

```

TEST=2 ' Respond to colors
GOSUB 2040

COLOR 7,0:CLS
SCREEN 1: COLOR 0,0
S=1: GOSUB 2650 ' Get start of session time

FOR T= 1 TO TRIALS
  'Check for end of session
  S=2:GOSUB 2650
  IF ZSEC(2) >= ZSEC(1) + 180 THEN NTC = T-1:GOTO 1910
  IF DIST(T)=1 THEN X=2:Y=0:STIMULUS=1:
    WORD=1:GOTO 1600 'RR
  IF DIST(T)=2 THEN X=2:Y=0:STIMULUS=1:
    WORD=2:GOTO 1600 'RG
  IF DIST(T)=3 THEN X=2:Y=0:STIMULUS=1:
    WORD=3:GOTO 1600 'RB
  IF DIST(T)=4 THEN X=1:Y=0:STIMULUS=2:
    WORD=1:GOTO 1600 'GR
  IF DIST(T)=5 THEN X=1:Y=0:STIMULUS=2:
    WORD=2:GOTO 1600 'GG
  IF DIST(T)=6 THEN X=1:Y=0:STIMULUS=2:
    WORD=3:GOTO 1600 'GB
  IF DIST(T)=7 THEN X=1:Y=1:STIMULUS=3:
    WORD=1:GOTO 1600 'BR
  IF DIST(T)=8 THEN X=1:Y=1:STIMULUS=3:
    WORD=2:GOTO 1600 'BG
  IF DIST(T)=9 THEN X=1:Y=1:STIMULUS=3:
    WORD=3:GOTO 1600 'BB
1600 IF TEST=1 OR TEST=3 THEN STIMULUS = WORD
    DAT(T,12)=WORD
    IF TEST=3 THEN Y=1: X=3 ' (All words are white)
1640 RS=INP(ADDRESS+13): IF RS<>255 THEN 1640
    ON WORD GOSUB 2190,2290,2400
1660 RS=INP(ADDRESS+13): IF RS=255 THEN 1690 ELSE 1660

    ' Key-1=red Key-2=green Key-4=blue
1690 K=0:GOSUB 2610
1700 S=2: RS=INP(ADDRESS+13)
    IF RS=255 THEN GOSUB 2650 ELSE GOTO 1740
    IF ZSEC(2) >= ZSEC(1)+180 THEN NTC=T-1:GOTO 1910
    GOTO 1700
1740 K=4:GOSUB 2610
    IF RS<>247 AND RS<>239 AND RS<>251 THEN GOTO 1700
    IF RS=247 THEN RS=1
    IF RS=239 THEN RS=2
    IF RS=251 THEN RS=3
    ANSWER=RS: DAT(T,10)=ANSWER
    IF ANSWER=4 THEN ANSWER=3
    IF STIMULUS=4 THEN STIMULUS=3
    DAT(T,11)=STIMULUS
    IF STIMULUS-ANSWER=0 THEN GOTO 1830

```

```

        ERRORS(T)=ERRORS(T) + 1 : DAT(T,9)=2
        GOTO 1840
1830    CORRECT(T)=CORRECT(T) + 1: DAT(T,9)=1
1840    CLS: OUT CRT,BLANK
        NEXT T

        NTC=TRIALS

1910 SCREEN 2:SCREEN 0:COLOR 7,0:CLS:LOCATE 1,1,0
        OUT ADDRESS+14,0
        COLUMNS=12
        OPEN "O",1,F$+".STP"
        PRINT#1,SN$;".CD$
        PRINT#1,NTC,COLUMNS,TEST
        FOR TR=1 TO NTC
            PRINT#1,DAT(TR,1);DAT(TR,2);DAT(TR,3);
            DAT(TR,4);DAT(TR,5);DAT(TR,6);
            PRINT#1,DAT(TR,7);DAT(TR,8);DAT(TR,9);
            DAT(TR,10);DAT(TR,11);DAT(TR,12)
        NEXT TR: CLOSE
        NXT=NXT+1
        IF NXT>N THEN PRINT "END OF SESSION":END:LOCATE 1,1,1
        CHAIN T$(NXT)

' Letters
2040 R$ = "R10U10R10F10R10H10R10U15L40D25BM
        +10, 15R20U5L20D5BM+50,15"
        E$ = "R40U5L30U5R30U5L30U5R30U5L40D25BM+60,0"
        D$ = "R35E5U15H5L35D25BM+10, 5R18E3U9H3L18D14BM+50,6"
        L$ = "R40U5L30U20L10D25BM+60,0"
        G$ = "R40U15L20D5R10D5L20U15R30U5L40D25BM+60,0"
        B$ = "R35E5U5H3E2U5H5L35D25BM
        +10, 5R20E1U3H1L20D5BM
        +0, 10R20E1U3H1L20D5BM+50,15"
        U$ = "R40U25L10D20L20U20L10D25BM+60,0"
        N$ = "R10U20F20R10U25L10D15H15L15D25BM+50,0"
        RED$=R$+E$+D$
        GREEN$=G$+R$+E$+E$+N$
        BLUE$=B$+L$+U$+E$
        RETURN

' Draw Red
2190 CLS: OUT CRT,BLANK
        COLOR 0,Y
        PSET (70,120),X
        DRAW RED$
        PAINT (73,112),X: PAINT (135,112),X: PAINT (193,112),X
        OUT CRT,UNBLANK
        RETURN

' Draw Green
2290 CLS: OUT CRT,BLANK

```

```

COLOR 0,Y
PSET (13,120),X
DRAW GREEN$
PAINT (15,110),X: PAINT (75,110),X: PAINT (135,110),X
PAINT (196,110),X: PAINT (258,110),X
OUT CRT,UNBLANK
RETURN
2380
' Draw Blue
2400 CLS: OUT CRT,BLANK
COLOR 0,Y
PSET (48,120),X
DRAW BLUE$
PAINT (50,110),X: PAINT (112,110),X
PAINT (173,110),X: PAINT (233,110),X
OUT CRT,UNBLANK
RETURN

2610 ' Read clock and store time in array
SETREGZ(AXREGZ)=&H2C00
CALL INT86(&H21,VARPTR(SETREGZ(0)),VARPTR(GETREGZ(0)))
HM=GETREGZ(CXREGZ): SH=GETREGZ(DXREGZ)
DAT(T,(K+1))=HM \ 256: DAT(T,(K+2))=HM MOD 256
DAT(T,(K+3))=SH \ 256: DAT(T,(K+4))=SH MOD 256
RETURN

2650 ' Convert Hr:Min:Sec to Sec and test for end
ZTIME$=TIME$
ZS=VAL(RIGHT$(ZTIME$,2))
ZM=VAL(MID$(ZTIME$,4,2))
ZH=VAL(LEFT$(ZTIME$,2))
ZM1=(ZH * 60) + ZM
ZSEC(S)=(ZM1 * 60) + ZS
RETURN

```

```

'          Grammatical Reasoning
'          GRAMPAN.BAS
'          07-29-88

' Generation of letter pairs and letter statements modified
'   from code by R. Irons and M. Krause, NBDL
REM $INCLUDE: 'COMMON.BAS'
ADDRESS=1808: OUT ADDRESS+15,146
OUT ADDRESS+14,0
SCREEN 2:SCREEN 0:CLS:LOCATE 1,1,0

'CRT=986:BLANK=0:UNBLANK=1:OUT 984,2  'Z-150
CRT=728:BLANK=7:UNBLANK=15           'COLOR 400-SH
DIM SETREG%(7),GETREG%(7)
AXREG%=0: CXREG%=2: DXREG%=3

DIM NT(33),Q(33),UC(33),VC(33),DAT(33,11),ZSEC(6)
DIM P$(2),C$(2),D$(2)
DIM U$(2),E$(2),F$(2)
DIM G$(2),H$(2),V$(2)

TT=0 ' Statement true and answer true
TF=0 ' Statement true and answer false
FF=0 ' Statement false and answer false
FT=0 ' Statement false and answer true

A$="A":B$="B"
V$(1)=" "
U$(1)=" "
V$(2)="S"
U$(2)=" DOES"
E$(1)=" "
E$(2)=" NOT"
F$(1)=" PRECEDE"
F$(2)=" FOLLOW"
G$(1)=" "
G$(2)="E"
H$(1)=" "
H$(2)="D BY"
P$(1)=" "+A$+" "+B$+" "
P$(2)=" "+B$+" "+A$+" "
C$(1)=" "+A$
C$(2)=" "+B$
D$(1)=" "
D$(2)=" IS"
TRIALS=32

```

```

LOCATE 7,10,0 ' Turn off cursor
GOSUB 2260
FOR X=1 TO 32
    IF Q(X)=2 OR Q(X)=3 OR Q(X)=6 OR Q(X)=7
        OR Q(X)=10 OR Q(X)=11 OR Q(X)=14 OR Q(X)=15 THEN
        UC(X)=1
    IF Q(X)=4 OR Q(X)=5 OR Q(X)=6
        OR Q(X)=7 OR Q(X)=12 OR Q(X)=13
        OR Q(X)=14 OR Q(X)=15 THEN VC(X)=1
NEXT X

'TIME ITI
S=5:GOSUB 2700
1581 S=6:GOSUB 2700
    IF ZSEC(6) < ZSEC(5)+14 THEN 1581
    COLOR 7,4:CLS 'RED SCREEN
1582 S=6:GOSUB 2700
    IF ZSEC(6) < ZSEC(5)+16 THEN 1582
    COLOR 7,14:CLS ' YELLOW SCREEN
1583 S=6:GOSUB 2700
    IF ZSEC(6) < ZSEC(5)+18 THEN 1583
    COLOR 7,2:CLS 'GREEN SCREEN
1584 S=6:GOSUB 2700
    IF ZSEC(6) < ZSEC(5)+20 THEN 1584

COLOR 7,1: OUT ADDRESS+14,4
I=0
S=1:GOSUB 2700          'Get start of session time

FOR RN=1 TO TRIALS
    'Check for end of session
    IF RN=1 THEN 1690
    S=2:GOSUB 2700
    IF ZSEC(2) >= ZSEC(1) + 180 THEN NTC=RN-1: GOTO 2130
1690 X=NT(RN)
    IF X>16 THEN I=2
    IF X<17 THEN I=1
    ' Print phrase
    CLS
    GOSUB 2510
    LOCATE 10,24
    'Store type of statement in data array
    IF K=1 AND L=1 AND M=2 THEN DAT(RN,10)=1:GOTO 1850
    IF K=1 AND L=1 AND M=1 THEN DAT(RN,10)=2:GOTO 1850
    IF K=2 AND L=1 AND M=2 THEN DAT(RN,10)=3:GOTO 1850
    IF K=2 AND L=1 AND M=1 THEN DAT(RN,10)=4:GOTO 1850
    IF K=1 AND L=2 AND M=2 THEN DAT(RN,10)=5:GOTO 1850
    IF K=1 AND L=2 AND M=1 THEN DAT(RN,10)=6:GOTO 1850
    IF K=2 AND L=2 AND M=2 THEN DAT(RN,10)=7:GOTO 1850
    IF K=2 AND L=2 AND M=1 THEN DAT(RN,10)=8
1850 'Store type of letter pairs in data array
    IF J=1 AND I=1 THEN DAT(RN,11)=1:GOTO 1900

```

```

IF J=1 AND I=2 THEN DAT(RN,11)=2:GOTO 1900
IF J=2 AND I=1 THEN DAT(RN,11)=3:GOTO 1900
IF J=2 AND I=2 THEN DAT(RN,11)=4

1900 PRINT $(J);D$(K);U$(KL);E$(L);F$(M);V$(LK);
      G$(KM);H$(K); C$(J1); " ";P$(I)

O=0:GOSUB 2660

' Get response and put in counter bin for right/wrong
S0=0:S1=0:S=2
IF ZSEC(2) >=ZSEC(1) + 180 THEN NTC=RN-1:GOTO 2130
2012 RS=INP(ADDRESS+13): IF RS<>255 THEN 2012
2013 RS=INP(ADDRESS+13)
IF RS<>247 AND RS<>251 THEN 2013
O=4:GOSUB 2660
IF RS=247 THEN S0=1
IF RS=251 THEN S1=1
IF B0=1 AND S0=1 THEN TT=TT+1:DAT(RN,9)=1
IF B1=1 AND S1=1 THEN FF=FF+1:DAT(RN,9)=1
IF B0=1 AND S1=1 THEN TF=TF+1:DAT(RN,9)=2
IF B1=1 AND S0=1 THEN FT=FT+1:DAT(RN,9)=2
NEXT RN

NTC=TRIALS
2130 SCREEN 2:SCREEN 0:COLOR 7,0:CLS:LOCATE 1,1,0
      COLUMNS=11: OUT ADDRESS+14,0
      COLOR 7,0:CLS
      OPEN "O",1,F$+".GRR"
      PRINT#1,SN$;" ";CD$
      PRINT#1,NTC,COLUMNS
      FOR TR=1 TO NTC
        PRINT#1,DAT(TR,1);DAT(TR,2);DAT(TR,3);
          DAT(TR,4);DAT(TR,5);DAT(TR,6);
          DAT(TR,7);DAT(TR,8);DAT(TR,9);
          DAT(TR,10);DAT(TR,11)
      NEXT TR:CLOSE
      NXT=NXT+1
      IF NXT>N THEN PRINT "END OF SESSION":END:LOCATE 1,1,1
      CHAIN T$(NXT)

2260 ' Routine for picking up random Q(1) through Q(32)
      FOR X 1 TO 16
2280 Q(X)=INT(16*RND)+1
          IF X 1 THEN 2330
          FOR NI 1 TO X 1
              IF Q(X)=Q(NI) THEN 2280
          NEXT NI
2330 NEXT X
      FOR X=17 TO 32
2350 Q(X)=INT(16*RND)+1
          IF X 17 THEN 2400

```



```

        FOR NI=17 TO X-1
            IF Q(X)=Q(NI) THEN 2350
        NEXT NI
2400 NEXT X
    ' Randomize NT
    FOR X=1 TO 32
2430     NT(X)=INT(32*RND)+1
        IF X=1 THEN 2480
        FOR NI=1 TO X-1
            IF NT(X)=NT(NI) THEN 2430
        NEXT NI
2480 NEXT X
    RETURN

2510 ' Set pointers for phrases
    FOR J=1 TO 2
        FOR K=1 TO 2
            FOR L=1 TO 2
                FOR M=1 TO 2
                    IF Q(X)=(8*J)+(4*K)+(2*L)+M-14 THEN 2550
                NEXT M
            NEXT L
        NEXT K
    NEXT J
2550 IF K*M=4 THEN KM=2 ELSE KM=1
    IF J=1 THEN J1=2 ELSE J1=1
    KL=1: LK=1
    IF K=1 AND L=2 THEN KL=2
    IF L+K=2 THEN LK=2
    SUM=I+J+K+L+M
    B0=0: B1=0
    IF SUM=5 OR SUM=7 OR SUM=9 THEN B0=1
    IF SUM=6 OR SUM=8 OR SUM=10 THEN B1=1
    RETURN

2660 'Read clock and store time in array
    SETREGZ(AXREGZ)=&H2C00
    CALL INT86(&H21,VARPTR(SETREGZ(0)),VARPTR(GETREGZ(0)))
    HM=GETREGZ(CXREGZ):SH=GETREGZ(DXREGZ)
    DAT(RN,(0+1))=HM \ 256: DAT(RN,(0+2))=HM MOD 256
    DAT(RN,(0+3))=SH \ 256: DAT(RN,(0+4))=SH MOD 256
    RETURN

2700 'Convert Hr:Min:Sec to Sec and test for end
    ZTIME$=TIME$
    ZS=VAL(RIGHT$(ZTIME$,2))
    ZM=VAL(MID$(ZTIME$,4,2))
    ZH=VAL(LEFT$(ZTIME$,2))
    ZM1=(ZH*60) + ZM
    ZSEC(S)=(ZM1*60) + ZS
    RETURN

```

```

'           Manikin
'           MANPAN.BAS
'           07-29-88

' The code for the figure outline is courtesy of
' D. Thorne, WRAIR
REM #INCLUDE: 'COMMON.BAS'
ADDRESS = 1808: OUT ADDRESS + 15, 146
OUT ADDRESS + 14, 0
SCREEN 2: SCREEN 0: CLS : LOCATE 1, 1, 0
'CRT=986: BLANK=0: UNBLANK=1: OUT 984,2      ' Z150
CRT = 728: BLANK = 7: UNBLANK = 15          ' Color 400-SH

DIM SETREGZ(7), GETREGZ(7)
AXREGZ = 0: CXREGZ = 2: DXREGZ = 3

DIM DAT(35, 12), ZSEC(6), SEQ(35), CORRECT(35), ERRORS(35)

' Random sequence for a block of 16 trials
FOR I = 1 TO 16
1110   A = INT(RND * 16 + 1)
       FOR J = 1 TO 16
           IF SEQ(J) = A THEN 1110
       NEXT J
       SEQ(I) = A
NEXT I

S = 5: GOSUB 4030
1130 S = 6: GOSUB 4030
      IF ZSEC(6) < ZSEC(5) + 14 THEN 1130 ' ITI Change screen
color R Y G
      COLOR 7, 4: CLS
1160 S = 6: GOSUB 4030
      IF ZSEC(6) < ZSEC(5) + 16 THEN 1160
      COLOR 7, 6: CLS
1190 S = 6: GOSUB 4030
      IF ZSEC(6) < ZSEC(5) + 18 THEN 1190
      COLOR 7, 2: CLS
1220 S = 6: GOSUB 4030
      IF ZSEC(6) < ZSEC(5) + 20 THEN 1220
      TRIALS = 32
      OUT ADDRESS + 14, 4
      TOTC = 0: TOTE = 0
      COLOR 7, 0: CLS
      S = 1: GOSUB 4030 ' Get start of session time

```

```

' Main Program
SCREEN 1: COLOR 0, 0
FOR I = 1 TO TRIALS
    S = 2: GOSUB 4030
    IF ZSEC(2) >= ZSEC(1) + 180 THEN NTC = I - 1: GOTO 1530
1370 RS = INP(ADDRESS + 13): IF RS <> 255 THEN 1370
    OUT CRT, BLANK
    A = SEQ((I MOD 16) + 1)
    ON A GOSUB 1680, 1720, 1760, 1800, 1840,
        1880, 1920, 1960, 2000, 2040, 2080, 2120,
        2160, 2200, 2240, 2280

    IF ANSWER = 1 THEN CORRECT(I) = CORRECT(I) + 1 ELSE
        ERRORS(I) = ERRORS(I) + 1
    IF ANSWER = 1 THEN TOTC = TOTC + 1 ELSE TOTE = TOTE + 1
    IF ANSWER = 1 THEN DAT(I, 9) = 1: GOTO 1460
    DAT(I, 9) = 2
1460 NEXT I

' End session
NTC = TRIALS
1530 SCREEN 2: SCREEN 0: COLOR 7, 0: CLS : LOCATE 1, 1, 0
    OUT ADDRESS + 14, 0
    OPEN "O", 1, F$ + ".MAN"
    COLUMNS = 11
    PRINT #1, SN$; ", "; CD$
    PRINT #1, NTC, COLUMNS
    FOR TR = 1 TO NTC
        PRINT #1, DAT(TR, 1); DAT(TR, 2); DAT(TR, 3);
        DAT(TR, 4); DAT(TR, 5); DAT(TR, 6);
        DAT(TR, 7); DAT(TR, 8); DAT(TR, 9);
        DAT(TR, 10); DAT(TR, 11)
    NEXT TR: CLOSE
    NXT = NXT + 1
    IF NXT > N THEN PRINT "END OF SESSION": END: LOCATE 1, 1, 1

    CHAIN T$(NXT)

' Subroutines:

' Body orient View Outline Shape-Shape location(r/l)

' Body-Front Circle-Circle on right
1680 DAT(I, 10) = 1: DAT(I, 11) = 1
    GOSUB 2310: GOSUB 2650: GOSUB 3310: GOSUB 3450: GOSUB 3640
    RETURN
' Body-Front Circle-Circle on left
1720 DAT(I, 10) = 1: DAT(I, 11) = 2
    GOSUB 2310: GOSUB 2650: GOSUB 3310: GOSUB 3370: GOSUB 3760
    RETURN
' Body-Front Square-Square on right
1760 DAT(I, 10) = 1: DAT(I, 11) = 3

```

```

GOSUB 2310: GOSUB 2650: GOSUB 3340: GOSUB 3370: GOSUB 3640
RETURN
' Body-Front-Square-Square on left
1800 DAT(I, 10) = 1: DAT(I, 11) = 4
GOSUB 2310: GOSUB 2650: GOSUB 3340: GOSUB 3450: GOSUB 3760
RETURN
' Body-Back-Circle-Circle on right
1840 DAT(I, 10) = 2: DAT(I, 11) = 1
GOSUB 2310: GOSUB 3150: GOSUB 3310: GOSUB 3450: GOSUB 3760
RETURN
' Body-Back-Circle-Circle on left
1880 DAT(I, 10) = 2: DAT(I, 11) = 2
GOSUB 2310: GOSUB 3150: GOSUB 3310: GOSUB 3370: GOSUB 3640
RETURN
' Body-Back-Square-Square on right
1920 DAT(I, 10) = 2: DAT(I, 11) = 3
GOSUB 2310: GOSUB 3150: GOSUB 3340: GOSUB 3370: GOSUB 3760
RETURN
' Body-Back-Square-Square on left
1960 DAT(I, 10) = 2: DAT(I, 11) = 4
GOSUB 2310: GOSUB 3150: GOSUB 3340: GOSUB 3450: GOSUB 3640
RETURN
' Reverse Revfront-Circle-Circle on right
2000 DAT(I, 10) = 3: DAT(I, 11) = 1
GOSUB 2820: GOSUB 2650: GOSUB 3310: GOSUB 3450: GOSUB 3760
RETURN
' Reverse-Revfront-Circle-Circle on left
2040 DAT(I, 10) = 3: DAT(I, 11) = 2
GOSUB 2820: GOSUB 2650: GOSUB 3310: GOSUB 3370: GOSUB 3640
RETURN
' Reverse Revfront-Square-Square on right
2080 DAT(I, 10) = 3: DAT(I, 11) = 3
GOSUB 2820: GOSUB 2650: GOSUB 3340: GOSUB 3370: GOSUB 3760
RETURN
' Reverse Revfront-Square-Square on left
2120 DAT(I, 10) = 3: DAT(I, 11) = 4
GOSUB 2820: GOSUB 2650: GOSUB 3340: GOSUB 3450: GOSUB 3640
RETURN
' Reverse-Back Circle-Circle on right
2160 DAT(I, 10) = 4: DAT(I, 11) = 1
GOSUB 2820: GOSUB 3150: GOSUB 3310: GOSUB 3450: GOSUB 3640
RETURN
' Reverse Back Circle Circle on left
2200 DAT(I, 10) = 4: DAT(I, 11) = 2
GOSUB 2820: GOSUB 3150: GOSUB 3310: GOSUB 3370: GOSUB 3760
RETURN
' Reverse Back Square Square on right
2240 DAT(I, 10) = 4: DAT(I, 11) = 3
GOSUB 2820: GOSUB 3150: GOSUB 3340: GOSUB 3370: GOSUB 3640
RETURN
' Reverse Back Square Square on left
2280 DAT(I, 10) = 4: DAT(I, 11) = 4

```

```

GOSUB 2820: GOSUB 3150: GOSUB 3340: GOSUB 3450: GOSUB 3760
RETURN
2310 ' Draw Outline
WINDOW SCREEN (0, 0)-(640, 325): CLS
' Draw body outline
2340 CIRCLE (320, 72), 20, 3, , , 1
LINE (363, 117)-(358, 124): LINE -(351, 162):
LINE -(355, 184): LINE -(398, 304)
LINE -(370, 310): LINE -(320, 195)
LINE (277, 117)-(283, 124): LINE -(289, 162):
LINE -(285, 184): LINE -(242, 304)
LINE -(270, 310): LINE -(320, 195)
LINE (331, 88)-(331, 93): LINE -(346, 99):
LINE -(372, 101): LINE -(490, 137)
LINE -(480, 150): LINE -(363, 119)
LINE (309, 88)-(309, 93): LINE -(294, 99): LINE -(268, 101):
LINE (150, 137)
LINE -(160, 150): LINE -(277, 119)
LINE (288, 162)-(351, 162): LINE (288, 164)-(352, 168), , B
RETURN
2650 ' Draw Front Markers
LINE (310, 60)-(339, 67): LINE (302, 68)-(313, 62):
PAINT (320, 60)
CIRCLE (320, 75), 21, 3, , , 1.1: PAINT (320, 90)
CIRCLE (312, 75), 4, , , , .4:
CIRCLE (329, 75), 4, , , , .4: PSET (312, 76), 1
PSET (329, 75), 1
CIRCLE (320, 78), 4, , 4, 5.4:
LINE (320, 78)-(320, 72): LINE (330, 70)
LINE (320, 72)-(310, 70): LINE (317, 85)-(324, 85), 2
LINE (331, 93)-(320, 97): LINE -(309, 93):
LINE (336, 95)-(328, 102)
LINE (320, 97): LINE -(312, 102): LINE (304, 95)
LINE (315, 100)-(318, 104): LINE (313, 155):
LINE (320, 161): LINE -(327, 155)
LINE (322, 104): LINE (325, 100): PAINT (320, 102), 3, 3:
PAINT (320, 158)
LINE (333, 118)-(351, 118): CIRCLE (320, 166), 4
LINE (317, 168)-(317, 192): LINE -(320, 195):
LINE -(323, 192): LINE -(323, 168)
RETURN
' Draw Reverse Body Outline
2820 WINDOW (0, 0)-(640, 325): CLS
GOSUB 2340
RETURN
3150 ' Draw Back Markers
PAINT (320, 72)
LINE (309, 92)-(331, 92): LINE (304, 96)-(336, 96)
LINE (300, 162)-(300, 170): LINE (340, 162)-(340, 170)
LINE (329, 174)-(347, 174): LINE -(347, 188):
LINE -(338, 191): LINE -(329, 188)
LINE -(329, 174): LINE (293, 174)-(311, 174):

```

```

LINE -(311, 188): LINE -(302, 191)
LINE -(293, 188): LINE -(293, 174)
LINE (320, 187)-(320, 195): LINE (320, 112)-(320, 142)
LINE (320, 194)-(332, 198): LINE -(345, 196)
LINE (320, 194)-(308, 198): LINE -(295, 196)
RETURN
3310 ' Draw Outer Circle
CIRCLE (320, 162), 239, 1
CIRCLE (320, 162), 231, 1
PAINT (320, 3), 1
RETURN
3340 ' Draw Outer Square
LINE (86, 1)-(554, 323), 2, B
LINE (95, 6)-(545, 318), 2, B
PAINT (320, 4), 2
RETURN
3370 ' Draw left circle and right square
CIRCLE (91, 162), 50, 3
PAINT (91, 162), 1, 3
LINE (505, 132)-(593, 192), 3, B
PAINT (515, 134), 2, 3
RETURN
3450 ' Draw left square and right circle
LINE (47, 132)-(135, 192), 3, B
PAINT (48, 134), 2, 3
CIRCLE (549, 162), 50, 3
PAINT (549, 162), 1, 3
RETURN

' Process key inputs
3640 OUT CRT, UNBLANK: RS = INP(ADDRESS + 13): IF RS <> 255 THEN
3640
3660 RS = INP(ADDRESS + 13): IF RS = 255 THEN 3670 ELSE 3660
3670 K = 0: GOSUB 4000 ' Get initial IRT value
3680 S = 2: RS = INP(ADDRESS + 13)
IF RS = 247 OR RS = 251 THEN GOTO 3720 ELSE GOSUB 4030
IF ZSEC(2) >= ZSEC(1) + 180 THEN NTC = 1 - 1: GOTO 1530
GOTO 3680
3720 K = 4: GOSUB 4000 ' Get time of response and store
IF RS = 247 THEN ANSWER = 1 ELSE ANSWER = 0
RETURN

3760 OUT CRT, UNBLANK: RS = INP(ADDRESS + 13):
IF RS <> 255 THEN 3760
3780 RS = INP(ADDRESS + 13): IF RS = 255 THEN 3790 ELSE 3780
3790 K = 0: GOSUB 4000 ' Get initial IRT value
3800 S = 2: RS = INP(ADDRESS + 13)
IF RS = 247 OR RS = 251 THEN GOTO 3840 ELSE GOSUB 4030
IF ZSEC(2) >= ZSEC(1) + 180 THEN NTC = 1 - 1: GOTO 1530
GOTO 3800
3840 K = 4: GOSUB 4000
IF RS = 251 THEN ANSWER = 1 ELSE ANSWER = 0

```

RETURN

```
4000 ' Read clock and store time in array
      SETREGX(AXREGX) = &H2C00
      CALL INT86(&H21, VARPTR(SETREGX(0)), VARPTR(GETREGX(0)))
      HM = GETREGX(CXREGX): SH = GETREGX(DXREGX)
      DAT(I, (K + 1)) = HM \ 256: DAT(I, (K + 2)) = HM MOD 256
      DAT(I, (K + 3)) = SH \ 256: DAT(I, (K + 4)) = SH MOD 256
      RETURN
```

```
4030 ' Convert Hr:Min:Sec to Sec and test for end
      ZTIME$ = TIME$
      ZS = VAL(RIGHT$(ZTIME$, 2))
      ZM = VAL(MID$(ZTIME$, 4, 2))
      ZH = VAL(LEFT$(ZTIME$, 2))
      ZM1 = (ZH * 60) + ZM
      ZSEC(S) = (ZM1 * 60) + ZS
      RETURN
```

```

'           Numerical Memory
'           NUMPAN.BAS
'           07-28-88

```

```

REM $INCLUDE: 'COMMON.BAS'
ADDRESS=1808: OUT ADDRESS+15,146
OUT ADDRESS+14,0
SCREEN 2:SCREEN 0:CLS:LOCATE 1,1,0
DIM SETREG%(7), GETREG%(7)
AXREG%=0: CXREG%=2: DXREG%=3

```

```

TRIALS=24
DIM TME(25),MXT(24,4),DAT(25,16),ZSEC(6),ZBCD(4),ZBD(4)
DIM ZTOTIME(4)
N1=0:N2=0:N3=0:N4=0:TRL=0:TA=0:V1=0:V2=0:V3=0:NTC=0
S=5:GOSUB 3810

```

```

' Assign 1-4 digit targets for each trial
FOR TRL=1 TO TRIALS
1190   TME(TRL)=INT(RND * 4) + 1
      IF TME(TRL) = 1 THEN N1=N1+1:
        IF N1 > TRIALS / 4 GOTO 1190
      IF TME(TRL) = 2 THEN N2=N2+1:
        IF N2 > TRIALS / 4 GOTO 1190
      IF TME(TRL) = 3 THEN N3=N3+1:
        IF N3 > TRIALS / 4 GOTO 1190
      IF TME(TRL) = 4 THEN N4=N4+1:
        IF N4 > TRIALS / 4 GOTO 1190
      DAT(TRL,1)=TME(TRL)
      FOR TA=1 TO TME(TRL) 'Assign digits to target
1260   MXT(TRL,TA) = INT (RND *10)
        IF TA=1 THEN GOTO 1360
        IF TA=3 THEN GOTO 1330
        IF TA=4 THEN GOTO 1350
        V1=MXT(TRL,TA): V3=TA-1: V2=MXT(TRL,V3)
        IF V1=V2 THEN GOTO 1260
        GOTO 1360
1330   IF MXT(TRL,3)=MXT(TRL,2) OR MXT(TRL,3)=MXT(TRL,1)
        THEN GOTO 1260
        GOTO 1360
1350   IF MXT(TRL,4)=MXT(TRL,3) OR MXT(TRL,4)=MXT(TRL,2)
        OR MXT(TRL,4)=MXT(TRL,1) THEN 1260
1360   NEXT TA
      DAT(TRL,12)=MXT(TRL,1): DAT(TRL,13)=MXT(TRL,2):
      DAT(TRL,14)=MXT(TRL,3): DAT(TRL,15)=MXT(TRL,4)
NEXT TRL

```



```

      ' ITI
1410 S=6:GOSUB 3810
      IF ZSEC(6) < ZSEC(5)+14 THEN 1410
      COLOR 7,4:CLS
1440 S=6:GOSUB 3810
      IF ZSEC(6) < ZSEC(5)+16 THEN 1440
      COLOR 7,6:CLS
1470 S=6:GOSUB 3810
      IF ZSEC(6) < ZSEC(5)+18 THEN 1470
      COLOR 7,2:CLS
1500 S=6:GOSUB 3810
      IF ZSEC(6) < ZSEC(5)+20 THEN 1500
      COLOR 7,0:CLS
      S=1:GOSUB 3810 'Start Time

SCREEN 1 ' Start session
OUT ADDRESS+14,4
FOR TRL=1 TO TRIALS
    COLOR 1,1
    S=2:GOSUB 3810 'Clock Time
    IF ZSEC(2) >= ZSEC(1)+180 THEN NTC=TRIALS-1:GOTO 1750
    CLS
1610    RS=INP(ADDRESS+13): IF RS(<>)255 THEN 1610
    GOSUB 1890 'Draw Sample
    S=1: GOSUB 4020
1645    S=2: GOSUB 4020:
        IF ZTotime(2) < ZTotime(1)+200 THEN 1645
        COLOR 0,0
        CLS: S=1: GOSUB 4020
1665    S=2: GOSUB 4020:
        IF ZTotime(2) < ZTotime(1)+300 THEN 1665
        GOSUB 2260 ' Probe Digit
        CLS
NEXT TRL

' End session
NTC=TRIALS
OUT ADDRESS+14,0
1750 SCREEN 2: SCREEN 0:COLOR 7,0:CLS:LOCATE 1,1,0
      OPEN "O",1,F#+" ".NUM"
      COLUMNS=16
      PRINT#1,SN$;" ",CD$
      PRINT#1,NTC,COLUMNS
      FOR TRL=1 TO NTC
          PRINT#1,DAT(TRL,1);DAT(TRL,2);DAT(TRL,3);
              DAT(TRL,4);DAT(TRL,5);DAT(TRL,6);
          PRINT#1,DAT(TRL,7);DAT(TRL,8);DAT(TRL,9);
              DAT(TRL,10);DAT(TRL,11);DAT(TRL,12);
          PRINT#1,DAT(TRL,13);DAT(TRL,14);DAT(TRL,15);
              DAT(TRL,16)
      NEXT TRL: CLOSE
      NXT-NXT+1

```

```

IF NXT>N THEN PRINT "END OF SESSION":END:LOCATE 1,1,1
CHAIN T$(NXT)

' Screen positions for digits
1890 ON TME(TRL) GOSUB 1920,1960,2030,2130
RETURN

1920 LOCATE 12,20
X=MXT(TRL,1):XP=X:PRINT XP
RETURN

1960 LOCATE 12,18
X=MXT(TRL,1):XP=X:PRINT XP;
M=MXT(TRL,2):XP=M:PRINT XP
RETURN

2030 LOCATE 12,17
X=MXT(TRL,1):XP=X:PRINT XP;
M=MXT(TRL,2):XP=M:PRINT XP;
XN=MXT(TRL,3):XP=XN:PRINT XP
RETURN

2130 LOCATE 12,15
X=MXT(TRL,1):XP=X:PRINT XP;
M=MXT(TRL,2):XP=M:PRINT XP;
XN=MXT(TRL,3):XP=XN:PRINT XP;
XZ=MXT(TRL,4):XP=XZ:PRINT XP
RETURN

2260 ' Choose if probe digit is new or an original target digit
LOCATE 12,20
W=INT(2*RND)+1
IF W=1 THEN DAT(TRL,10)=1:GOSUB 2530
IF W=2 THEN DAT(TRL,10)=2:GOSUB 2860
2310 RS=INP(ADDRESS+13): IF RS <> 255 THEN 2310

' Check for response
2350 RS=INP(ADDRESS+13): IF RS=255 THEN 2360 ELSE 2350
2360 J=0:GOSUB 3770 ' Initial IRT value
2370 S=2: RS=INP(ADDRESS+13)
IF RS=255 THEN GOSUB 3810 ELSE GOTO 2410
IF ZSEC(2) > ZSEC(1)+180 THEN NTC=TRIALS-1: GOTO 1750
GOTO 2370
2410 J=4:GOSUB 3770
IF RS<>247 AND RS<> 251 THEN GOTO 2370
IF RS=247 THEN RS=1
IF RS=251 THEN RS=3
F=0
FOR TA= 1 TO TME(TRL)
IF MXT(TRL,TA)=P THEN F=1
NEXT TA
IF F=1 AND RS=1 THEN DAT(TRL,9)=1:GOTO 2500

```

```

        IF F=0 AND RS=3 THEN DAT(TRL,9)=1:GOTO 2500
        DAT(TRL,9)=2
2500 RETURN

        ' Pick which digit will reappear
2530 ON TME(TRL) GOSUB 2570,2620,2690,2770
        RETURN

2570 P=MXT(TRL,1): XP=P
        PRINT XP
        DAT(TRL,16)=XP
        RETURN

2620 Q= INT(2 * RND) + 1
        IF Q=1 THEN P=MXT(TRL,1)
        IF Q=2 THEN P=MXT(TRL,2)
        XP=P: PRINT XP
        DAT(TRL,16)=XP
        RETURN

2690 Q= INT (3* RND) +1
        IF Q=1 THEN P=MXT(TRL,1)
        IF Q=2 THEN P=MXT(TRL,2)
        IF Q=3 THEN P=MXT(TRL,3)
        XP=P: PRINT XP
        DAT(TRL,16)=XP
        RETURN

2770 Q = INT (4 * RND) +1
        IF Q=1 THEN P=MXT(TRL,1)
        IF Q=2 THEN P=MXT(TRL,2)
        IF Q=3 THEN P=MXT(TRL,3)
        IF Q=4 THEN P=MXT(TRL,4)
        XP=P: PRINT XP
        DAT(TRL,16)=XP
        RETURN

2860 ON TME(TRL) GOSUB 2900,2960,3020,3080
        RETURN

2900 P= INT (10 * RND)
        IF P=MXT(TRL,1) THEN GOTO 2900
        XP=P: PRINT XP
        DAT(TRL,16)=XP
        RETURN

2960 P= INT (10 * RND)
        IF P=MXT(TRL,1) OR P=MXT(TRL,2) THEN GOTO 2960
        XP=P: PRINT XP
        DAT(TRL,16)=XP
        RETURN

```

```

3020 P= INT (10 * RND)
      IF P=MXT(TRL,1) OR P=MXT(TRL,2) OR P=MXT(TRL,3)
          THEN GOTO 3020
      XP=P: PRINT XP
      DAT(TRL,16)=XP
      RETURN

3080 P= INT (10 * RND)
      IF P=MXT(TRL,1) OR P=MXT(TRL,2) OR P=MXT(TRL,3)
          OR P=MXT(TRL,4) THEN 3080
      XP=P: PRINT XP
      DAT(TRL,16)=XP
      RETURN

3770 ' Read clock
      SETREGZ(AXREGZ)=&H2C00
      CALL INT86(&H21,VARPTR(SETREGZ(0)),VARPTR(GETREGZ(0)))
      HM=GETREGZ(CXREGZ): SH=GETREGZ(DXREGZ)
      DAT(TRL,(J+1))=HM \ 256: DAT(TRL,(J+2))=HM MOD 256
      DAT(TRL,(J+3))=SH \ 256: DAT(TRL,(J+4))=SH MOD 256
      RETURN

3810 ' Convert HR:MIN:SEC to seconds and test for end
      ZTIME$=TIME$
      ZS=VAL(RIGHT$(ZTIME$,2))
      ZM=VAL(MID$(ZTIME$,4,2))
      ZH=VAL(LEFT$(ZTIME$,2))
      ZM1=(ZH*60)+ZM
      ZSEC(S)=(ZM1*60)+ZS
      RETURN

4020 ' Time interval duration
      SETREGZ(AXREGZ)=&H2C00
      CALL INT86(&H21,VARPTR(SETREGZ(0)),VARPTR(GETREGZ(0)))
      HM=GETREGZ(CXREGZ): SH=GETREGZ(DXREGZ)
      Z1=HM \ 256: ZTOH=Z1*360000
      Z2=HM MOD 256: ZTOM=Z2*6000
      Z3=SH \ 256: ZTOS=Z3*100
      ZTOHS=SH MOD 256
      ZTOTIME(S)=ZTOH+ZTOM+ZTOS+ZTOHS
      RETURN

```

```

'           Simultaneous Pattern Comparison
'           PATPAN.BAS
'           07-28-88

' The algorithms for generating the X and Y dot coordinates
' are derived from R. Irons and P. Rose, NBDL
REM $INCLUDE: 'COMMON.BAS'
ADDRESS=1808: OUT ADDRESS+15,146
OUT ADDRESS+14,0
SCREEN 2:SCREEN 0:CLS:LOCATE 1,1,0
DIM X1(8),Y1(8),X2(8),Y2(8),ZSEC(6),DAT(60,10),ZTIME(2)

'CRT=986:BLANK=0:UNBLANK=1:OUT 984,2 'Z-150
CRT=728:BLANK=7:UNBLANK=15 'Color 400-SH
DIM SETREGZ(7),GETREGZ(7)
AXREGZ=0:CKREGZ=2:DXREGZ=3

LOCATE 1,1,0

' Time inter-test-interval
S=5:GOSUB 2540
1130 S=6:GOSUB 2540
IF ZSEC(6) < ZSEC(5)+14 THEN 1130
COLOR 7,4:CLS
1160 S=6:GOSUB 2540
IF ZSEC(6) < ZSEC(5)+16 THEN 1160
COLOR 7,6:CLS
1190 S=6:GOSUB 2540
IF ZSEC(6) < ZSEC(5)+18 THEN 1190
COLOR 7,2:CLS
1220 S=6:GOSUB 2540
IF ZSEC(6) < ZSEC(5)+20 THEN 1220
TRIALS=60
S=1:GOSUB 2540 ' Get start of session time
SCREEN 1:COLOR ,1 :CLS: OUT ADDRESS+14,4

FOR TR=1 TO TRIALS
S=2:GOSUB 2540:IF ZSEC(2) >= ZSEC(1)+180
THEN NTC=TR 1:GOTO 2280
' Choose X coordinate for eight dots
FOR I=1 TO 8
1350 X1(I)=(INT((RND(1)*139)+10)) ' Yields # between 10
and 149
X2(I)=X1(I)
FOR TD=1 TO I
IF X1(I)=X1(I-TD) or X1(I)=X1(I-TD)+3 or
X1(I)=X1(I-TD)-3 THEN 1350

```

```

        NEXT TD
    NEXT I

    ' Choose Y coordinates for eight dots
    FOR J=1 TO 8
1440      Y1(J)=(INT((RND(1)*160)+15)): ' Yields # between 15
        and 175
        Y2(J)=Y1(J)
        FOR TD=1 TO J
            IF Y1(J)=Y1(J-TD) or Y1(J)=Y1(J-TD)+3 or
                Y1(J)=Y1(J-TD)-3 THEN 1440
        NEXT TD
    NEXT J

    ' Generate different dot
1540      B=(INT((RND(1)*8)+1))
1550      X2(B)=(INT((RND(1)*139)+10))
        FOR C=1 TO 8
            IF C=B THEN 1590
            IF X2(B)=X2(C) THEN 1550
1590      NEXT C
1600      Y2(B)=(INT((RND(1)*160)+15))
        FOR D=1 TO 8
            IF D=B THEN 1640
            IF Y2(B)=Y2(D) THEN 1600
1640      NEXT D
        IF X2(B) <= (X1(B)+10) AND X2(B) >= (X1(B)-10) THEN
            1660 ELSE 1680
1660      IF Y2(B) <= (Y1(B)+10) AND Y2(B) >= (Y1(B)-10) THEN
            1540

1680 ' Choose whether pattern two is same or different
        A=INT(RND(1)*2)
        IF A<1 THEN DAT(TR,10)=1:GOTO 1740 ' Same
        IF A=1 THEN DAT(TR,10)=2:GOTO 2000 ' Different

1740 ' Draw same patterns
        GOSUB 2420 ' Blank screen
        CLS
        LINE (1,1)-(318,190),1,B
        LINE (159,1)-(159,190),1
        FOR I=1 TO 8
            X1=X1(I):Y1=Y1(I)
            X2=X1(I)+159:Y2=Y1(I)
            LINE (X1,Y1)-(X1+1,Y1+1),1,B
            LINE (X2,Y2)-(X2+1,Y2+1),1,B
        NEXT I
        GOSUB 2480 ' Unblank screen
1880      RS=INP(ADDRESS+13):IF RS<>255 THEN 1880
        J=0:GOSUB 2630 ' Initial IRT value
1900      S=2
        RS=INP(ADDRESS+13):IF RS=247 OR RS=251 THEN 1950

```

```

ELSE GOSUB 2540
IF ZSEC(2) >= ZSEC(1)+180 THEN NTC=TR-1:GOTO 2280
GOTO 1900
1950 J=4:GOSUB 2630 ' Second IRT value
IF RS=251 THEN DAT(TR,9)=2:GOTO 1980 ' Incorrect
response
IF RS=247 THEN DAT(TR,9)=1 ' Correct response
1980 GOTO 2230

2000 ' Draw different patterns
GOSUB 2420
CLS
LINE (1,1)-(318,190),1,B
LINE (159,1)-(159,190),1
FOR I=1 TO 8
X1=X1(I):Y1=Y1(I)
X2=X2(I)+159:Y2=Y2(I)
LINE (X1,Y1)-(X1+1,Y1+1),1,B
LINE (X2,Y2)-(X2+1,Y2+1),1,B
NEXT I
GOSUB 2480
2140 RS=INP(ADDRESS+13):IF RS <> 255 THEN 2140
J 0:GOSUB 2630 ' Initial IRT value
2160 S=2
RS=INP(ADDRESS+13):IF RS=247 OR RS=251 THEN 2200
ELSE GOSUB 2540
IF ZSEC(2) >= ZSEC(1)+180 THEN NTC=TR-1:GOTO 2280
GOTO 2160
2200 J=4:GOSUB 2630 ' Second IRT value
IF RS=247 THEN DAT(TR,9)=2:GOTO 2230 ' Incorrect
response
IF RS=251 THEN DAT(TR,9)=1 ' Correct response
2220
2230 GOSUB 2670 ' ITI
NEXT TR

' Write data to disk
NTC=TRIALS
2280 SCREEN 2:SCREEN 0:COLOR 7,0:CLS:LOCATE 1,1,0
OUT ADDRESS+14,0
OPEN "O",1,F$+"PAT"
COLUMNS 10
PRINT#1,SN$;" ";CD$
PRINT#1,NTC,COLUMNS
FOR TR=1 TO NTC
PRINT#1,DAT(TR,1);DAT(TR,2);DAT(TR,3);DAT(TR,4);
DAT(TR,5);DAT(TR,6);DAT(TR,7);DAT(TR,8);
DAT(TR,9);DAT(TR,10)
NEXT TR:CLOSE
NXT=NXT+1
IF NXT>N THEN PRINT "END OF SESSION":END:LOCATE 1,1,1
CHAIN T$(NXT)

```

```

2420 ' Toggle blank
      OUT CRT,BLANK
      RETURN

2480 ' Toggle unblank
      OUT CRT,UNBLANK
      RETURN

2540 ' Convert HR:MIN:SEC to seconds
      ZTIME$=TIME$
      ZS=VAL(RIGHT$(ZTIME$,2))
      ZM=VAL(MID$(ZTIME$,4,2))
      ZH=VAL(LEFT$(ZTIME$,2))
      ZM1=(ZH*60) + ZM
      ZSEC(S)=(ZM1*60) + ZS
      RETURN

2630 ' Read clock
      SETREGZ(AXREGZ)=&H2C00
      CALL INT86(&H21,VARPTR(SETREGZ(0)),VARPTR(GETREGZ(0)))
      HM=GETREGZ(CXREGZ): SH=GETREGZ(DXREGZ)
      DAT(TR,(J+1))=HM \ 256: DAT(TR,(J+2))=HM MOD 256
      DAT(TR,(J+3))=SH \ 256: DAT(TR,(J+4))=SH MOD 256
      RETURN

2670 ' Time inter-trial-interval
      CLS
      S=1:GOSUB 2700
2680 S=2:GOSUB 2700:IF ZTIME(2) < ZTIME(1)+25 THEN 2680
      RETURN

2700 ' Clock values in hundreds of seconds
      SETREGZ(AXREGZ)=&H2C00
      CALL INT86(&H21,VARPTR(SETREGZ(0)),VARPTR(GETREGZ(0)))
      HM=GETREGZ(CXREGZ): SH=GETREGZ(DXREGZ)
      Z1 HM \ 256: ZH=Z1*360000
      Z2 HM MOD 256: ZM=Z2*6000
      Z3 SH \ 256: ZE=Z3*100
      ZHS SH MOD 256
      ZTIME(S)=ZH+ZM+ZE+ZHS
      RETURN

```



```

'           Repeated Acquisition
'           RAPAN.BAS
'           07-29-88

```

```

REM $INCLUDE: 'COMMON.BAS'
ADDRESS=1808: OUT ADDRESS+15,146
OUT ADDRESS+14,0
SCREEN 2:SCREEN 0:CLS:LOCATE 1,1,0
S=5:GOSUB 2100 'Start inter-test-interval

```

```

'CRT=986:BLANK=0:UNBLANK=1:OUT 984,2      'Z-150
CRT=728:BLANK=7:UNBLANK=15                'Color 400-SH
DIM SETREGZ(7),GETREGZ(7)
AXREGZ=0: CXREGZ=2: DXREGZ=3

```

```

DEFINT A-Y
DIM ZTOTIME(4)
DIM C$(16),C(16),DAT(1000,10),ZSEC(6),ZBD(4),
    ZBCD(4),SEQUENCE$(24)
SEQS=0:TO1=3:TO2=48:CORRECT=0:ERRORS=0:R=1
TRIALS=25:LENGTH=12

```

```

' Read sequences from DATA statements
FOR I=1 TO 24
    READ SEQUENCE$(I)
NEXT I
CHAINN$=SEQUENCE$(SEQNUM)

```

```

' Convert "response sequence" to integer
FOR I=1 TO LENGTH
    C$(I) MID$(CHAINN$,I,1)
    C(I)=VAL (C$(I))
NEXT I

```

```

' Time inter test interval
1310 S=6:GOSUB 2100
    IF ZSEC(6) < ZSEC(5)+14 THEN 1310
    COLOR 7,4:CLS ' Screen red
1340 S=6:GOSUB 2100
    IF ZSEC(6) < ZSEC(5)+16 THEN 1340
    COLOR 7,6:CLS ' Screen yellow
1370 S=6:GOSUB 2100
    IF ZSEC(6) < ZSEC(5)+18 THEN 1370
    COLOR 7,2:CLS ' Screen green
1400 S=6:GOSUB 2100
    IF ZSEC(6) < ZSEC(5)+20 THEN 1400
    CLS:SCREEN 1:COLOR 0,0

```

```

S=1:GOSUB 2090 ' Get start time

OUT ADDRESS+14,4
FOR I=1 TO TRIALS
  ' Generate stimulus display (rectangles)
  X1=14:Y1=6:X2=67:Y2=54:X=15:Y=13:BOX=0
  CLS
  S=2:GOSUB 2090
  IF ZSEC(2) >= ZSEC(1)+300 THEN 1845
  FOR J=1 TO 3
    FOR K=1 TO 4
      LINE (X1,Y1)-(X2,Y2),3,B
      X1=X1+80:X2=X2+80
      BOX=BOX+1:IF BOX=LENGTH THEN 1620
    NEXT K
    X1=14:X2=67:Y1=Y1+60:Y2=Y2+60
  NEXT J
1620  X1=14:Y1=6:X2=67:Y2=54:K=0:J=0:B=0

  ' Check response
  FOR M=1 TO LENGTH
1660    RS=INP(ADDRESS+13):IF RS<>255 THEN 1660
1670    H=0:GOSUB 2050 'Beginning IRT value
1680    S=2
    RS=INP(ADDRESS+13):IF RS=255 THEN GOSUB 2090
    ELSE 1735
    IF ZSEC(2) >= ZSEC(1)+300 THEN 1845
    GOTO 1680
1735    IF RS=247 OR RS=239 OR RS=251 THEN 1742 ELSE 1680
1742    H=4:IF RS=247 THEN RS=1
    IF RS=239 THEN RS=2
    IF RS=251 THEN RS=3
    GOSUB 2050
    IF RS=C(M) THEN 2380 ELSE 2470 ' Correct or
    incorrect response
1760  NEXT M
  FOR M1=1 TO 10:DAT(R,M1)=0:NEXT M1 ' Zero's
  indicate sequence completion
  R=R+1
  S=5:GOSUB 2100
1792  S=6:GOSUB 2100:IF ZSEC(6) < ZSEC(5)+1 THEN 1792
  NEXT I

  GOSUB 2320 ' Unblank screen
1845  SCREEN 2:SCREEN 0:COLOR 7,0:CLS:LOCATE 1,1,0
  OUT ADDRESS+14,0

  ' Write data to disk
  NTC=R-1
  OPEN "O",1,F$+".ACQ"
  COLUMNS 10
  PRINT #1,SN$;","CD$;","CHAINN$

```

```

PRINT #1,NTC,COLUMNS,LENGTH
FOR TR=1 TO NTC
    PRINT#1,DAT(TR,1);DAT(TR,2);DAT(TR,3);DAT(TR,4);
    DAT(TR,5);DAT(TR,6);DAT(TR,7);DAT(TR,8);
    DAT(TR,9);DAT(TR,10)
NEXT TR:CLOSE
NXT=NXT+1
IF NXT>N THEN PRINT "END OF SESSION":END:LOCATE 1,1,1
CHAIN T$(NXT)

1970 ' Paint squares
LINE (X1,Y1)-(X2,Y2),2,BF
X1=X1+80:X2=X2+80
J=J+1:B=B+1
IF J=4 THEN 2020 ELSE 2030
2020 X1=14:X2=67:Y1=Y1+60:Y2=Y2+60:J=0
2030 RETURN

2050 ' Read clock and store time in data array
SETREGZ(AXREGZ)=&H2C00
CALL INT86(&H21,VARPTR(SETREGZ(0)),VARPTR(GETREGZ(0)))
HM=GETREGZ(CXREGZ): SH=GETREGZ(DXREGZ)
DAT(R,(H+1))=HM \ 256: DAT(R,(H+2))=HM MOD 256
DAT(R,(H+3))=SH \ 256: DAT(R,(H+4))=SH MOD 256
RETURN

2090 ' Convert HR:MIN:SEC to seconds
2100 ZTIME$=TIME$
ZS=VAL(RIGHT$(ZTIME$,2))
ZM=VAL(MID$(ZTIME$,4,2))
ZH=VAL(LEFT$(ZTIME$,2))
ZM1=(ZH*60)+ZM
ZSEC(S)=(ZM1*60)+ZS
RETURN

2180 ' Timeout routine - blank screen
GOSUB 2260 ' Toggle blank screen
S=1:GOSUB 2700
2210 S=2:GOSUB 2700:IF ZTOTIME(2) < ZTOTIME(1)+100 THEN 2210
GOSUB 2320 ' Toggle unblank screen
RETURN

2260 ' Toggle blank screen
OUT CRT,BLANK
RETURN

2320 ' Toggle unblank
OUT CRT,UNBLANK
RETURN

2380 ' Correct response
GOSUB 1970

```

```

S=3:GOSUB 2700
DAT(R,9)=C(M) ' Response code
DAT(R,10)=M ' Sequence position
R=R+1:CORRECT=CORRECT+1
2430 S=4:GOSUB 2700:IF ZTotime(4) < ZTotime(3)+3 THEN 2430
GOTO 1760

2470 ' Incorrect response
IF RS=1 THEN DAT(R,9)=4:GOTO 2510
IF RS=2 THEN DAT(R,9)=5:GOTO 2510
IF RS=3 THEN DAT(R,9)=6
2510 DAT(R,10)=M
R=R+1:ERRORS=ERRORS+1
GOSUB 2180
GOTO 1670

DATA 1312132313232121,2323212131231313,
1231312323123212,1312323132321213
DATA 3121232312132312,1312312121323231,
3231321312321213,3132132313212121
DATA 2132313132312121,3213132312312123,
2321212132313132,2312323121312131
DATA 3212321213213132,1232132123213131,
1313121213232323,2313231232121313
DATA 3232132121321312,2123123213132312,
3132321313212121,1323231312312123
DATA 2323212313131212,3212121313212313,
2323231312123131,3123213121323123

2700 ' Timeout duration
SETREGZ(AXREGZ)=&H2C00
CALL INT86(&H21,VARPTR(SETREGZ(0)),VARPTR(GETREGZ(0)))
HM=GETREGZ(CXREGZ):SH=GETREGZ(DXREGZ)
Z1=HM \ 256:ZTOH=Z1*360000
Z2=HM MOD 256:ZTOM=Z2*6000
Z3=SH \ 256:ZTOS=Z3*100
ZTOHS=SH MOD 256
ZTotime(S)=ZTOH+ZTOM+ZTOS+ZTOHS
RETURN

```

```

      Visual Scanning
      VISPAN.BAS
      07-29-88

```

```

REM #INCLUDE: 'COMMON.BAS'
ADDRESS=1808: OUT ADDRESS+15,146
OUT ADDRESS+14,0
SCREEN 2:SCREEN 0:CLS:LOCATE 1,1,0
DIM A(26),DAT(25,12),ZSEC(6)
DIM SETREGZ(7),GETREGZ(7)
AXREGZ=0: CXREGZ=2: DXREGZ=3

S=5:GOSUB 2150
1120 S=6:GOSUB 2150
      IF ZSEC(6) < ZSEC(5)+14 THEN 1120
      COLOR 7,4:CLS
1150 S=6:GOSUB 2150
      IF ZSEC(6) < ZSEC(5)+16 THEN 1150
      COLOR 7,14:CLS
1180 S=6:GOSUB 2150
      IF ZSEC(6) < ZSEC(5)+18 THEN 1180
      COLOR 7,2:CLS
1210 S=6:GOSUB 2150
      IF ZSEC(6) < ZSEC(5)+20 THEN 1210
      TRL=24 ' number of trials
      NUM=20 ' number of letters in row
      LETT=2 'number of target letters
      COLOR 7,0:CLS
      OUT ADDRESS+14,4
      S=1:GOSUB 2150          ' Get start of session time

      FOR I=1 TO TRL
        S=2:GOSUB 2150
        IF ZSEC(2) >= ZSEC(1) + 180 THEN NTC=I-1:GOTO 1990
        FOR J=1 TO 26: A(J)= J + 64 : NEXT J

        ' Choose letters for sequence
        FOR J= 1 TO 26: T=INT (RND (1) * (26-J)) + J
          U=A(T) :A(T)=A(J) :A(J)=U
        NEXT J
        C$=""
        FOR J= 1 TO NUM: C$ = C$ + " " + CHR$(A(J)):NEXT J
        Y= INT (RND (1) * 2) + 1
        ON Y GOTO 1450,1610

1450 ' Pick target digits from letter sequence (same condition)
      DAT(I,10)=1

```

```

D$=""
FOR J= 1 TO NUM
    T= INT (RND (1) * (NUM-J) ) + J
    U=A(T) : A(T)=A(J) : A(J)=U
NEXT J
FOR J= 1 TO LETT
    Q= INT (RND (1) * (NUM-J)) + J
    D$= D$ + " " + CHR$(A(Q))
    IF J=1 THEN DAT(I,11)= A(Q)
    IF J=2 THEN DAT(I,12)=A(Q)
    A(Q) = A(J)
NEXT J
AN= 247 :GOTO 1790

1610 ' Pick new target letters (different conditionN)
    DAT(I,10)=2
    FOR J= 1 TO NUM
        Q= INT (RND (1) * (NUM+1-J)) + J
        U= A(Q) : A(Q)= A(J) : A(J) = U
    NEXT J
    T= NUM-INT (RND (1) * LETT)
    D$=""
    FOR J= 1 TO LETT
        Q= INT (RND (1) * (LETT-J)) + J
        D$= D$ + " " + CHR$(A(T+Q))
        IF J=1 THEN DAT(I,11) = A(T+Q)
        IF J=2 THEN DAT(I,12)=A(T+Q)
        A(T+Q)= A(T+J)
    NEXT J
    AN=251

' Print target and sequence letters
1790 RS=INP(ADDRESS+13):IF RS<>255 THEN 1790
    LOCATE 5,39 :COLOR 6,0 :PRINT D$
    LOCATE 8,20 :PRINT C$:COLOR 7,0
1820 RS=INP(ADDRESS+13):IF RS=255 THEN 1830 ELSE 1820
1830 K=0:GOSUB 2110 ' Get initial IRT value

' Check for subject's response
1860 S=2:RS=INP(ADDRESS+13)
    IF RS=255 THEN GOSUB 2150 ELSE 1900
    IF ZSEC(2) >= ZSEC(1) + 180 THEN NTC=1 : GOTO 1990
    GOTO 1860
1900 K=4: GOSUB 2110
    IF RS<>247 AND RS<>251 THEN 1860
    IF RS=AN THEN 1925 ELSE 1930
1925 DAT(I,9)=1:GOTO 1940
1930 DAT(I,9)=2
1940 CLS
NEXT I

NTC=TRL

```

```

1990 SCREEN 2:SCREEN 0:COLOR 7,0:CLS:LOCATE 1,1,0
    OUT ADDRESS+14,0
    OPEN "O",1,F#+"VIS"
    COLUMNS=12
    PRINT #1,SN#;"",CD#
    PRINT #1,NTC,COLUMNS
2040 FOR TR=1 TO NTC
    PRINT#1,DAT(TR,1);DAT(TR,2);DAT(TR,3);DAT(TR,4);
        DAT(TR,5);DAT(TR,6);DAT(TR,7);DAT(TR,8);DAT(TR,9);
        DAT(TR,10);DAT(TR,11);DAT(TR,12)
    NEXT TR:CLOSE
    NXT=NXT+1
    IF NXT>N THEN PRINT "END OF SESSION":END:LOCATE 1,1,1
    CHAIN T$(NXT)

2110 ' Read clock and storetime in array
    SETREGZ(AXREGZ)=&H2C00
    CALL INT86(&H21,VARPTR(SETREGZ(0)),VARPTR(GETREGZ(0)))
    HM=GETREGZ(CXREGZ):SH=GETREGZ(DXREGZ)
    DAT(I,(K+1))=HM \ 256: DAT(I,(K+2))=HM MOD 256
    DAT(I,(K+3))=SH \ 256: DAT(I,(K+4))=SH MOD 256
    RETURN

2150 ' Convert HR:MIN:SEC to seconds
    ZTIME$=TIME$
    ZS=VAL(RIGHT$(ZTIME$,2))
    ZM=VAL(MID$(ZTIME$,4,2))
    ZH=VAL(LEFT$(ZTIME$,2))
    ZM1=(ZH*60) + ZM
    ZSEC(S)=(ZM1*60) + ZS
    RETURN

```

```
'      Common
'      COMMON.BAS
'      07-28-88
```

```
DEFINT A-Y ' Defined as integer for speed in execution
DIM T$(20)
COMMON SN$,CD$,NXT,T$(),N,F$,SEQNUM
```

```
' SN$ = subject name
' CD$ = current date
' NXT = next test in sequence
' T$() = string array containing sequence of tests
' N = number of tests in battery
' F$ = string variable for file name
' SEQNUM = variable containing number of repeated
'      acquisition sequence
```